Pre-Activations Research for Hysteresis Activation Function: Project Proposal Winter 2024-2025

Idan Kashani - idan-kashani@campus.technion.ac.il
Moshe Kimhi - moshekimhi@campus.technion.ac.il

Background

Activation functions are essential components of neural networks, introducing the non-linearities needed to model complex patterns in data. In recent years, several activation functions have become popular across both natural language processing (NLP) and computer vision (CV) tasks. Among these, the Gaussian Error Linear Unit (GELU) and similar fucntions, such as SiLU (Sigmoid Linear Unit) and Swish, have demonstrated remarkable performance and became a common standard in many vision and language models. These functions are often praised for their smoothness and performance, but their computational complexity poses challenges, especially in real-time and resource-constrained environments.
To address these limitations, this project proposes the development of an alternative activation function, HeLU, that is designed for improved efficiency with a minor sacrifice of performance. Unlike GELU and similar variants, HeLU aims to reduce computational overhead during training and inference while maintaining the desired model behavior.
We define the Hysteresis Rectifier Linear Unit (HeLU) as:

Definition 1 (HeLU) $HeLU_\alpha(x) = ReLU(x) = \max\{0, x\}$

Whereas, we override the autograd derivative that is used during backpropagation, and use instead the following term, lightly abusing the notation:

Definition 2 (HeLU Modified Derivative) $\frac{d}{dx}\ HeLU_\alpha(x) =$

$$\left\{ \begin{array}{ll} 0 & x \leq -\alpha \\ 1 & -\alpha < x \end{array} \right\}$$

Shifting back the derivative, essentially, refines the trigger of "dying ReLU", by demanding a greater step to turn off the neuron. For convenience, we added a simple implementation of HeLU in Torch-like pseudo code in Algorithm 1.

Algorithm 1 HeLU Function in PyTorch Style

```
class HeLUFunction(torch.autograd.Function)
@Staticmethod
def forward(ctx, z, alpha)
  ReLU ← torch.where(z > 0, z, 0)
  ctx.save_for_backward(z)
  ctx.alpha ← alpha
  return ReLU
@Staticmethod
def backward(ctx, grad_output)
  z, ← ctx.saved_tensors
  alpha ← ctx.alpha
  grad_positive ← torch.ones_like(z)
  grad_HeLU ← torch.where(z > -alpha, grad_positive, 0)
  return grad_HeLU * grad_output
```

Project Proposal

This project offers an opportunity to take part in the research of HeLU, a novel activation function that could serve as an efficient alternative to GELU. Through hands-on research, students will collect real-time statistics during neural network training, focusing on key elements such as pre-activation and gradients distributions. The project will investigate how these statistics relate to the optimal functioning of HeLU, with the goal of refining its implementation for both language and vision tasks. By the end of the project, students will gain valuable experience in evaluating activation functions and optimizing them based on real-time training data. Building on previous findings, this project aims to extend those results, with the potential for conference submission upon successful completion.

Objectives

- **Understand Activation Functions**: Study the theory and mathematical properties of popular activation functions like GELU, SiLU, and Swish, focusing on their advantages and limitations in different neural architectures.
- **Gather Training-Time Statistics**: Use PyTorch to collect and analyze real-time statistics related to pre-activations and gradients during model training.
- **Research the optimal connection**: Using the statistics, find the optimal setting of HeLU.
- **Gain Hands-On Experience**: Learn to conduct experiments, including

training and evaluation, using track results using tools like Weights & Biases (wandb), enhancing research skills in deep learning.

Eligibility and Timeline

Undergraduate students who have completed a courese in deep learning (CS236781, EE046211 or similar) can take the project. It offers a unique opportunity to gain practical research experience in a fast-evolving field, with potential for publishing findings in a conference.
For additional information, please contact us at the emails provided.

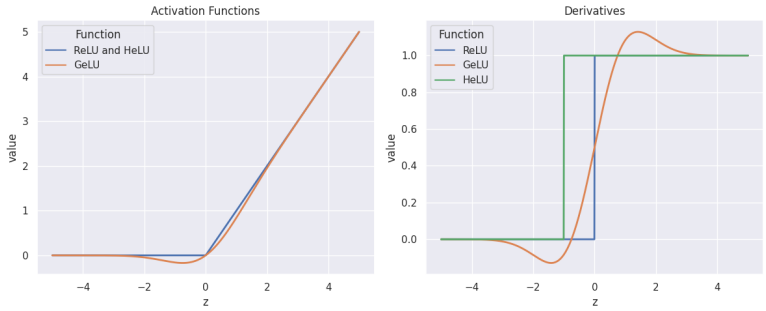

Figure 1: ReLU, GELU and HeLU functions and their derivatives

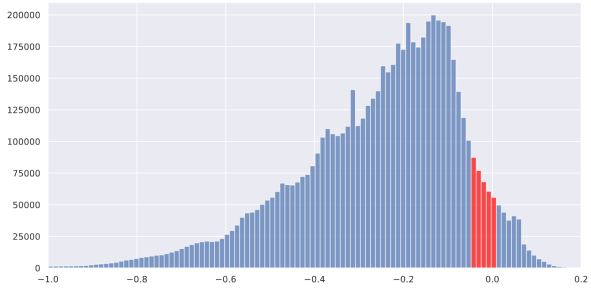

Figure 2: Weight distribution of all pre activations of Wide ResNet 40-4. In the image classification task with over parametric network, we see that majority of the pre activation features are negative. In red we mark the activations that we allow regret with HeLU.